基于改进花朵授粉的 K-均值聚类算法 *

陶志勇 [1]，刘晓芳 [1, 2†]，刘　影 [1]，王和章 [1]

(1. 辽宁工程技术大学 电子与信息工程学院，辽宁 葫芦岛 125105; 2. 阜新力兴科技有限责任公司，辽宁 阜新 123000)

摘　要：针对 K-means 聚类算法依赖于初始值并易陷入局部最优值的问题，提出了一种基于改进花朵授粉的 K-means 聚类算法。该算法首先通过混沌映射的序列作为花朵种群的初值位置，保证花朵种群在搜索空间的多样性、确定性；然后在花朵授粉的后期搜索阶段引入禁忌搜索算法以避免陷入局部最优解；最后将改进后的 FPA 算法用以优化 K-means 算法的初值。在 5 个聚类数据集上的实验结果表明，改进后算法的平均聚类准确率相比于花朵授粉聚类算法提高了 12.2%，证明了该算法对于低维数据集具有更好的聚类效果。

关键词：聚类；花朵授粉；混沌映射；禁忌搜索；k-means

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2018.05.0301

K-means clustering algorithm based on improved flower pollination

Tao Zhiyong[1], Liu Xiaofang[1, 2†], Liu Ying[1], Wang Hezhang[1]

(1. *School of Electronic & Information Engineering Liaoning Technical University*, *Huludao Liaoning* 125105, *China* 2. *Fuxin LiXing Technology Company Limited*, *Fuxin Liaoning* 123000, *China*)

Abstract: In order to solve the problem that k-means clustering algorithm is dependent on the initial value and easily falls into the local optimum, this paper proposed a K-means clustering algorithm based on improved flower pollination. Firstly, the algorithm used the chaotic map sequence as the initial position of the flower population to ensure the diversity and determinacy of the flower population in the search space; Then, it introduced a tabu search algorithm in the late stage of flower pollination to avoid falling into the local optimal solution; Finally, used the improved flower pollination algorithm to optimize the initial value of the k-means algorithm. Experimental results on five clustering datasets show that the improved algorithm improves the average clustering accuracy by 12.2% compared with the flower pollination clustering algorithm, which proves that the proposed algorithm has better clustering performance for low-dimensional datasets.

Key words: clustering; flower pollination; logistic map; tabu search; K-means

0　引言

在当前应用于计算机数据分析的各种方法中聚类是较为常见且广泛应用的算法，尤其适合于进行模式识别、数据挖掘、图像分析等工作[1]，这种算法分析的结果会依据数据的相似度对结果分类，使得相似度高的数据划为一类，相似度低的数据区分到不同种类。

K-means 因其实现简单、收敛速度快的优点而成为常用的聚类算法，但其聚类效果易受初始聚类中心的影响，而随机选取聚类中心的方法易使算法陷入局部最优值，为此许多学者对 K-means 算法初始聚类中心的选取进行了改进。Duwairi 等人[2]针对传统 K-means 算法对初值中心敏感的问题，提出了一种初始化球形 K-means 算法，该方法在初始化阶段随机扰动未知解，并为集群的紧凑性引入了一种新的评估度量方法，该度量方法测量矢量对应聚类中心的方向离散度，并根据离散度确定最终聚类结果。Kumar 等人[3]将聚类中心定位在数据集的高密度区并确保每个聚类中心相隔较远，而密度区通过 kd 树进行标志，此方法提高了 K-means 算法的聚类性能，但增加了算法的时间复杂度。Bianchi 等人[4]提出了一种基于密度的非度量空间，它使用输入数据中最具有代表性的模型估计聚类中心，该方法不依赖数据集的形状，但算法的计算量十分庞大。为此，学者考虑使用群智能算法优化初始聚类中心。如 Hu 等人[5]将差分进

收稿日期：2018-05-19；**修回日期：**2018-07-03　**基金项目：**辽宁省博士启动基金资助项目（20170520098）；辽宁省自然科学基金资助项目（2015020100）；辽宁省普通高等教育本科教学改革研究项目（551610001095）；辽宁省教育厅一般项目（LJ2017QL013）

作者简介：陶志勇（1978-），男，辽宁葫芦岛人，副教授，博士，主要研究方向为多媒体通信；刘晓芳（1995-），女（通信作者），辽宁鞍山人，硕士研究生，主要研究方向为机器学习（1367251096@qq.com）；刘影（1983-），女，吉林大安人，讲师，博士，主要研究方向为无线定位、物联网.王和章（1992-），男，河北邢台人，硕士研究生，主要研究方向为无线传感器定位.

化算法中的差向量代替果蝇算法的随机搜索，结合后的算法用于优化 K-means 算法。Rahman 等人[6]提出将遗传算法与 K-means 结合用于自动确定聚类中心，但因遗传算法局部搜索能力弱并易早熟，导致聚类精度不高。王波等人[7]提出了自适应布谷鸟与 K-means 相结合的算法，并利用 MapReduce 编程模型实现了算法的并行化，但因布谷鸟算法自身的缺点导致最终的聚类效果不理想且运行时间长。

花朵授粉算法（flower pollination algorithm, FPA）[8]是最新提出的一种群智能优化算法，基础的花朵授粉算法由于某些缺点已被学者们进行了改进，其中国外学者 Draa 对此进行了定性和定量分析[9]；Sayed 提出了将克隆技术和花朵授粉算法融合在一起的混合二进制算法，并以此进行特征选择 [10]；Galvez 提出了一种多模态的花朵授粉算法，通过多模式功能对原始花朵授粉算法进行增强，以便在优化问题中找到所有可能的最优解[11]。国内学者对 FPA 算法也进行了大量的改进，文献[12]将差分进化策略与 FPA 算法进行融合，增强了种群的多样性，提高了算法的全局搜索能力；文献[13]在算法的全局寻优阶段采用自适应步长策略，在局部寻优阶段引入单纯形法以提高搜索能力；文献[14]利用高斯变异对全局搜索进行扰动以提高种群多样性，并加入了 Powell 法以提高局部开发能力。虽然以上文献对 FPA 算法的寻优能力进行了改进，但其仍存在收敛速度慢、寻优精度低的缺点。

基于此，本文提出一种结合混沌理论和禁忌搜索的花朵授粉算法。该算法首先利用混沌理论初始化花朵种群，增加种群的多样性，加快算法的迭代速度；其次在搜索后期引入禁忌搜索，避免算法陷入局部最优解；然后将改进后的花朵授粉算法应用于 K-means 算法上，以确定聚类中心，增强了聚类效果。

1 聚类算法和花朵授粉算法

1.1 聚类相关问题

聚类问题是指：对于一个样本数为 n 的数据集 $Y=\{Y_1,Y_2,\cdots,Y_n\}$，将其划分为 k 个类，即 $C=\{C_1,C_2,\cdots,C_k\}$。数据聚类的目的是尽量减小数据间的距离，即减小数据点和它所属集合的中心(C_j)，表达式为

$$MSE=\sum_{i=1}^{n}\min\{\|Y_i-C_j\| \mid j=1,2,\cdots,k\} \tag{1}$$

其中：Y_i 是给定的数据集，C_j 是聚类中心。

1.2 K-means 算法

传统的 K-means 算法是从数据集随机选择 k 个数据点作为初始聚类中心，对于剩下的点分配给离其最近的聚类中心，然后将每一类的平均值作为新的聚类中心，循环这一过程。K-means 算法的过程如下：

a)随机选择 k 个点作为聚类中心。

b)确定剩余数据点到其最近的聚类中心。

c)重新计算聚类中心。

d)重复这一过程直到每个数据点和最近的数据中心的平方和不变。

1.3 花朵授粉算法

花朵授粉算法是模拟大自然中开花植物授粉的群智能优化算法。花朵授粉的过程分为两种，一种为异花授粉，其中异花授粉在大自然中是指授粉过程需要借助外力，比如蜜蜂、昆虫等授粉者且其符合莱维飞行，这一过程在花朵授粉算法中称为全局搜索。自花授粉是指花粉的传播不需要授粉者，而是利用风进行授粉，这一过程为局部搜索。算法中的全局搜索和局部搜索由转换概率 p 决定。在现实中，每朵花可产生数百万乃至更多的花粉，为了简化问题，在算法中假设每颗显花植物仅有一朵花每朵花仅有一个花粉，这意味着一朵花或一个花粉对应优化问题中的一个解。

花朵授粉算法需要达到以下理想条件：

a）生物异花授粉过程中携带花粉的传播者（鸟、蜜蜂等）通过莱维飞行进行全局授粉；

b）非生物自花授粉是指算法中的局部搜索过程；

c）花的常性是指繁衍概率，繁衍概率与参与的两朵花的相似性成比例关系；

d）转换概率 $p\in[0,1]$ 决定全局搜索和局部搜索之间的转换，由于风和物理距离等其他因素的影响，在整个授粉过程中，p 值的选取非常关键；

因此，以上理想条件在花朵授粉算法可以用数学公式进行描述。当 $p>rand$ 时，算法执行全局授粉，可由式（2）实现。

$$x_i^{t+1}=x_i+\gamma L(\lambda)(g_*-x_i^t) \tag{2}$$

其中：x_i^{t+1}、x_i^t 分别指第 $t+1$ 代和第 t 代的解，g_* 表示当前种群中的最优解，γ 是控制步长的缩放因子，本文中 $\gamma=1$，$L(\lambda)$ 表示对应于花朵个体的莱维飞行位移，$L(\lambda)$ 的表示式如下：

$$L(\lambda)\sim\frac{\lambda\Gamma(\lambda)\sin(\frac{\lambda}{2}\pi)}{\pi}\frac{1}{s^{1+\lambda}}(s>>s_0>0) \tag{3}$$

其中：$\Gamma(\lambda)$ 为伽马函数、$\lambda=3/2$，s 由式（4）决定。

$$s=\frac{\mu}{|v|^{1/\lambda}};\mu\sim N(0,\sigma^2),v\sim N(0,1) \tag{4}$$

式中的 σ^2 由式（5）得到：

$$\sigma^2=\left(\frac{\Gamma(1+\lambda)}{\lambda\Gamma[\frac{1+\lambda}{2}]}\cdot\frac{\sin(\frac{\lambda}{2}\pi)}{2^{\frac{\lambda-1}{2}}}\right)^{\frac{1}{\lambda}} \tag{5}$$

当 $p<rand$ 时，算法进行局部授粉，如式（6）所示。

$$x_i^{t+1}=x_i^t+\varepsilon(x_j^k-x_i^k) \tag{6}$$

其中：$\varepsilon\in[0,1]$，x_i^t 是第 t 代花粉 i，解向量 x_i、x_j^t、x_k^t 分别代表同种植物的不同花朵的花粉，等同于种群的两个随机解，可增强种群的多样性，从而提高算法的局部搜索能力。

2 改进的花朵授粉算法

基本的花朵授粉算法存在如下两个缺点：a)种群缺乏多样性；b)收敛速度慢，易陷入局部最优。针对缺点 a）本文引用混

沌序列来增强多样性；针对缺点 b）本文在后期搜索阶段引用禁忌搜索表，使花粉以较快速度找到最优解。

2.1 混沌优化策略

花朵授粉算法如常见的启发式优化算法一样对初始值比较敏感，因其在初始化阶段使用随机初始化，导致寻优过程中会增加迭代次数，尤其在处理复杂的非线性和多模态的问题时会降低整体的速度和精度。而混沌序列因其自身的特性可以弥补随机初始化种群分布不理想的缺陷。

混沌序列已经被应用于大量的演化算法，同时也证明该理论对于增加种群多样性、提高算法的收敛速度具有可行性。混沌序列因本身具有遍历性，相比于盲目无序的随机搜索利用混沌变量进行优化搜索更有优越性，同时可减小演化算法陷入局部最优解的缺点。其基本原理是将待优化的未知量映射到混沌空间（[0,1]），利用混沌映射规则，在混沌空间中搜索，并将搜索后得到的解映射回原始空间。用混沌序列初始化种群，可以使花朵个体在解空间进行遍历搜索，克服原始花朵种群初始化分布不均匀的问题。有多种生成混沌序列的混沌映射，本文使用 Logistic map 来产生混沌序列的初始花朵种群，其函数的形式如式（7）所示[15]。

$$x_{i+1} = \alpha x_i (1 + x_i) \tag{7}$$

其中: α 是混沌系数， $\alpha \in [0,4]$ ， 本文取 $\alpha = 4$ 。

在式（7）中， x_{i+1} 是随机初始化的混沌序列， x_{i+1} 的范围在 0~1。使用混沌序列初始化花朵种群的过程为：首先随机初始化生成一个具有 n 个花粉的种群 P；然后使用式（7）产生与种群 P 对应的混沌种群 CP。由于此方法对于使用混沌方式得到的种群仍然有初始化的个体，为了减小这部分个体对整体种群造成降低求解精度、减慢收敛速度的影响，对每次迭代得到的最优解采用式(7)进行一次混沌映射，得到的混沌解与最优解对比，若混沌解较优，则使用混沌解替代当前最优解，否则，保留当前的最优解。

2.2 禁忌搜索法

禁忌搜索算法（tabu search，TS）是一个用于局部优化的启发式算法，与常见的局部优化算法不同的是该算法采用禁忌技术，即禁止重复前面的工作，使用一个禁忌表记录下已经到达过的局部最优点，在下一次搜索中，利用禁忌表中的信息不再或有选择地搜索这些点[16]。上述过程使用流程框图描述如图 1 所示。采用的停止条件:给定每次运行后总循环的次数，即最大迭代步数。

3 基于改进花朵授粉的 K-均值聚类算法

基于以上两点改进提出了基于改进花朵授粉的 k 均值聚类算法。该算法的基本思想是：通过改进的 FPA 算法进行一次迭代寻优，将得到的新位置作为 K-means 算法的初始点并进行一次聚类，再用聚类获得的新的中心点更新花群，反复交替执行 FPA 算法和 K-means 算法直至算法结束。

改进后的 FPA 算法计算步骤描述如下：

a)对所有参数进行初始化。

b)对种群 P 中的 n 个花粉使用 Logistic Map，利用式（1）计算每个花粉 $\{z_1, z_2, \cdots, z_n\}$ 的适应度 $f(x)$ 。

c)对初始花群进行一次 K-均值聚类，再次利用式（1）对每个花粉的适应度值进行计算，记录当前所得到的全局最优解和其对应的最优值。

d)如若 $p > rand$ ， 则利用式（2）对所得到的解进行更新，并对解进行越界处理。

e)如若 $p < rand$ ，那么根据式（6）对解进行更新，并对解进行越界处理。

f)将 d)e)得到新解的适应度值与未更新的解进行比较，若新解的适应度更优，则用新解替换未更新的解作为最优解，否则保留未更新的解和其适应度值。对新花粉进行一次 K-means 聚类，并用划分后形成的新聚类中心更新花粉。

g)若 $t > [N_iter / 2]$ 转至 Step8， 否则转至 Step4。

h)利用 TS 算法的基本步骤对新种群进行局部寻优。

i)判断结束条件，若满足，则输出聚类结果；否则转至 d)。

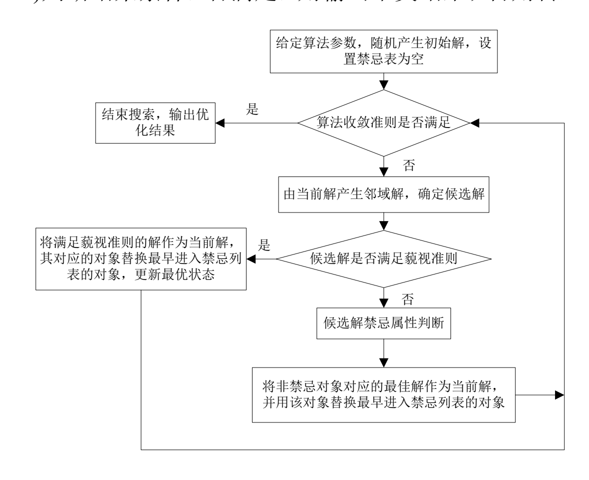


图 1 禁忌搜索流程

4 仿真实验与结果分析

为了验证本文算法的优越性及有效性，采用两组实验对改进算法进行验证，第一组实验是对本文算法、文献[12]算法（DEFPA）、标准 FPA 算法、差分进化算法（differential evolution，DE）算法、粒子群算法（particle swarm optimizati- on，PSO）算法及人工蜂群算法（artificial bee colony，ABC）分别在 5 个数据集上进行算法有效性测试；第二组实验是对数据集的聚类能力进行测试。实验环境：CPU 为 Inter Core i3-2350，内存为 4 GB，操作系统为 Window 7，开发软件为 Matlab2015a。

4.1 改进 FPA 算法性能测试

在本节中，选取了 5 个数据集来验证所提出算法的有效性，其中包括 2 个人工数据集和 3 个实际数据集（从 UCI 机器库中选取）。对于这些数据集，每个算法分别运行 20 次，得到的最优值、最差值、平均值和标准偏差分别记录在表 1、2、4、5、6 中，其中粗体表示本文算法优于另两个算法，下划线表示其他算法更好。针对每个数据集的算法收敛曲线如图 2~6 所示。

其中人工数据集 1（art1）是一个 3 维 5 类包含 250 个样本点的数据集，每一类服从均值分布，分别为 $U_1(85,100)$，$U_2(70,85)$，$U_3(55,70)$，$U_4(40,55)$，$U_5(25,40)$ [17]；表 1 列出了 art1 数据集的算法比较，算法的收敛曲线如图 2 所示。

表 1 在 art1 上各算法适应度比较

算法	最优值	最差值	均值	标准偏差
本文算法	**1709.31**	**2105.73**	1984.45	189.98
DEFPA	1806.67	2109.28	1978.65	86.02
FPA	1977.74	2418.71	2173.41	115.49
DE	1752.89	2495.90	1992.03	200.67
PSO	1773.82	2444.89	2205.41	312.73
ABC	1902.56	2206.78	2079.42	107.49

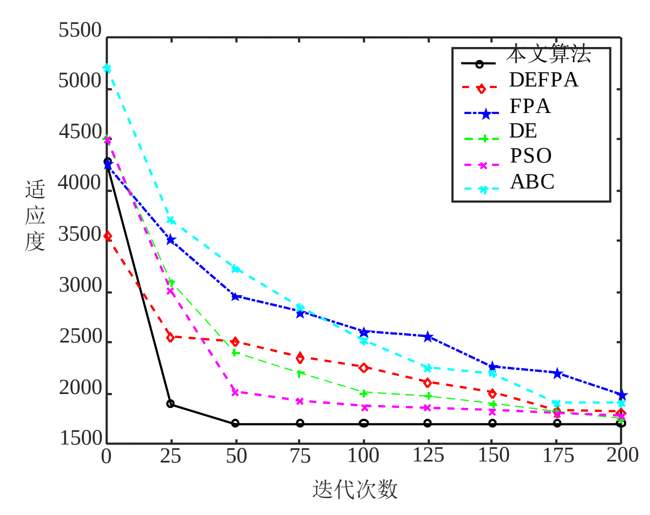


图 2 各算法在 art1 上的收敛图

由表 1 可知，本文算法的最优值、最差值均优于其余 5 个算法，虽然 DEFPA 算法的均值和标准偏差较小，但其最优值较差。同时由图 2 的 6 条仿真曲线可知，与其他 5 个算法相比，本文算法的适应度曲线较平滑，且收敛速度更快。这主要是由于本文算法在初始阶段加入混沌策略增强了全局搜索能力，加快了收敛速度，同时禁忌列表的引入保证了算法避开局部最优区域。虽然相比 DEFPA 算法在迭代初期聚类效果没有较大程度的改善，但在 25 次迭代之内本文算法有找到最优解的趋势。相比于 DEFPA 算法和 DE 算法，PSO 算法的平均适应度略大，但在迭代 50 次能较快地趋于最优解，收敛速度更快，ABC 算法和 FPA 算法随机初始种群，导致在 75 次迭代之前表现较差，寻找最优解能力较差。

人工数据集 2（art2）是一个 2 维 4 类包含 600 个样本点的数据集，所有的数据点由 4 个独立的双变量正态分布组成，分布形式如式（8）所示。

$$N_2\left(\mu=\begin{pmatrix} m_i \\ 0 \end{pmatrix}, \Sigma=\begin{bmatrix} 0.5 & 0.05 \\ 0.05 & 0.5 \end{bmatrix}\right) \quad (8)$$

$$i=1,2,3,4, m_1=-3, m_2=0, m_3=3, m_4=6$$

其中：μ 和 Σ 分别表示均值向量和协方差矩阵[17]。对于人工数据集 2 的算法比较如表 2 所示，算法收敛情况如图 3 所示。

表 2 在 art2 上各算法适应度比较

算法	最优值	最差值	均值	标准偏差
本文算法	**512.05**	**512.07**	**512.06**	**2.13e-12**
DEFPA	513.67	515.45	514.07	0.52
FPA	517.18	568.86	538.64	14.96
DE	512.01	514.21	513.94	0.08
PSO	513.90	514.20	513.95	0.04
ABC	514.69	534.42	518.32	4.43

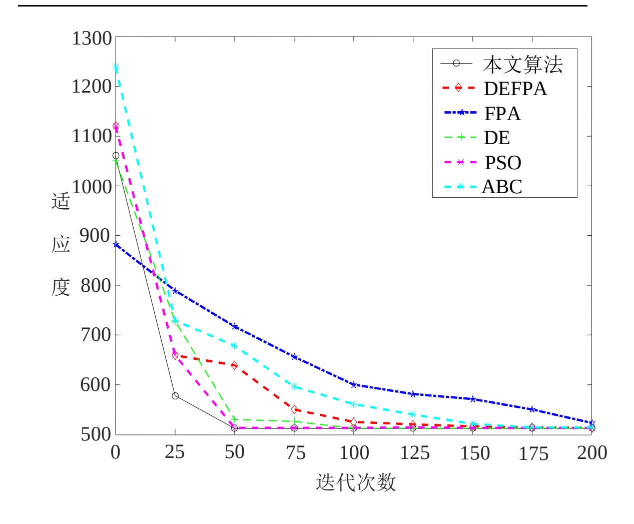


图 3 各算法在 art2 上的收敛图

由表 2 可知，本文算法的最优值、最差值、均值、标准差在 6 个算法中最好，其中适应度的大小表示聚类效果的好坏，适应度越小表示聚类效果越好，本文算法的最优适应度低于 FPA 算法 5.13，且标准差接近于 0，表明本文算法的聚类效果明显，稳定性较好。由图 3 可知，本文算法在 25 次迭代之内以较快速度接近全局最优解，在 25~50 次迭代之内即达到最优解，这表明本文算法的全局和局部搜索能力有明显增强，能精准确定聚类中心，提高聚类效果。PSO 算法在 50 次迭代达到的最优值与本文算法相近，但因局部搜索能力较弱，其易陷入局部最优。DEFPA 算法在 25~50 次迭代之间适应度变化不大，表明已陷入局部最优。ABC 算法的初始阶段随机选择聚类中心导致前期聚类效果较差，且在局部最优解附近多次迭代导致收敛速度减慢。FPA 算法初始阶段表现良好，但寻找聚类中心的能力较弱，导致聚类适应度最高。

从 UCI 中选取的 3 个真实数据集[18]的各项属性如 3 表所示。

表 3 数据集属性

数据集	样本点	维数	类别数
Iris	150	4	3
Wine	178	13	3
Heart	270	13	2

其中 3 个数据集的算法比较值在表 4~6 中，收敛情况如图 4~6 所示。

表 4 在 Iris 上的算法适应度比较

算法	最优值	最差值	均值	标准偏差
本文算法	**94.56**	**94.56**	**94.56**	**7.8e-13**
DEFPA	96.67	97.09	96.34	0.23
FPA	97.47	103.57	99.79	1.68
DE	96.65	105.85	97.57	2.00
PSO	96.65	127.66	105.96	14.57
ABC	97.14	100.29	98.10	0.68

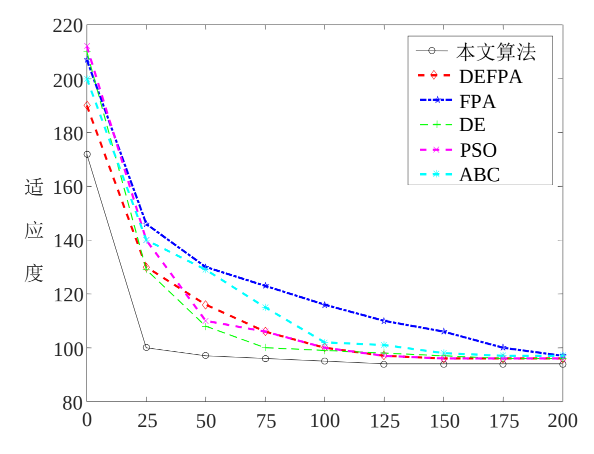


图 4 各算法在 Iris 上的收敛图

从表 4 可以看出，本文算法的最优值、最差值、均值、标准偏差均优于其余 5 个算法，适应度均值与 PSO 算法低 11，表明引入 Logistic 映射和禁忌表的 FPA 算法可准确确定 k-means 聚类中心，提高聚类精度。结合图 4 可知，在初期阶段，由于将解映射到混沌区间，增加了未知解搜索区域，使得算法的适应度变化较快。在搜索后期，由于禁忌表记录已搜索过的解，使算法不局限于局部空间的开采能力，避免了局部最优解。DE、DEFPA、PSO、ABC 算法可达到相同的最优解，PSO 前期表现较差，但收敛速度极快。FPA 算法的收敛速度较缓慢，所达到的最优解也非最优。

表 5 在 wine 上的算法适应度比较

算法	最优值	最差值	均值	标准偏差
本文算法	**16293.67**	**16294.37**	**16293.76**	**0.87**
DEFPA	16296.48	16306.91	16299.75	2.76
FPA	16322.06	16381.60	16343.73	16.14
DE	16336.35	18124.03	16876.52	512.39
PSO	16293.89	16297.61	16294.22	1.19
ABC	16391.45	17439.25	16706.49	249.14

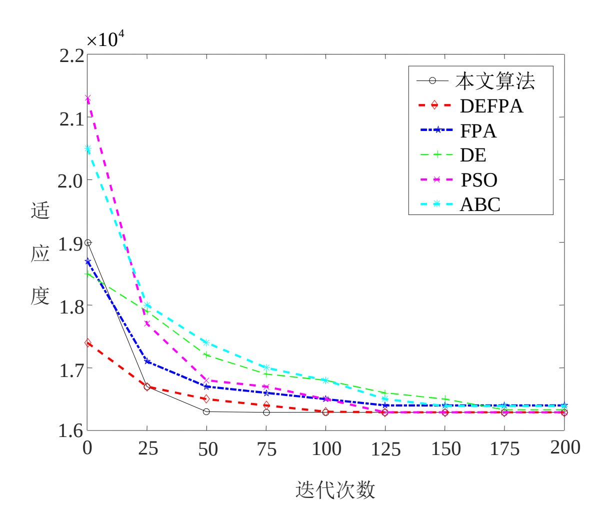


图 5 各算法在 wine 上的收敛图

从表 5 可知，在 wine 数据集下，本文算法的最优值、最差值、均值、标准偏差均优于其他算法，标准差低于 DE 算法 511.52，表明本文算法每次迭代得到的适应度相差较小，聚类稳定性更好。由图 5 可知，在迭代初期，本文算法适应度相比 DEFPA 算法更大，但在第 25 次迭代可与 DEFPA 算法达到相同的适应值，后期的适应值更小于 DEFPA 算法，且在迭代后期最先收敛到最优值。虽然 DEFPA 算法的最优解与本文算法一致，但是迭代速度较慢。FPA 算法总体状态优于 PSO、DE、ABC 算法，DE、ABC 虽然可得到较好最优解，但收敛极慢，PSO 算法可以较快速度收敛到最优解。

由表 6 可知，本文算法的最优值、最差值、均值、标准偏差均最好，最优值低于 ABC 算法 49.56，表明本文算法可准确的确定聚类中心，达到稳定聚类效果。从图 6 可看出，DE 算法与本文算法的收敛情况均较好，在 25 次迭代之内本文算法即接近最优解，而 DE 算法则在 25~50 次迭代之间收敛缓慢，未能避开局部最优区域。其主要原因是 DE 算法对参数设置敏感，不适当的参数会导致局部最优问题，而本文算法的禁忌表存储已搜索过的局部解，避免了多次搜索从而跳出局部区域。DEFPA、PSO 算法初期下降更快，但离最优解较远，还需多次迭代，FPA 算法与本文算法有相近的初始适应度，但因全局勘测和局部开采能力较弱，使得收敛速度缓慢，ABC 算法因其易早熟且局部寻优能力较弱导致适应度偏高且不易达到最优值。

表 6 在 Heart 上的适应度比较

算法	最优值	最差值	均值	标准偏差
本文算法	**10623.93**	**10624.79**	**10623.78**	**0.02**
DEFPA	10624.28	10624.98	10624.78	0.52
FPA	10630.40	10646.41	1063.98	4.01
DE	10628.37	11283.58	10739.11	173.61
PSO	10624.20	10624.91	10624.67	0.11
ABC	10673.49	11148.80	10795.84	110.15

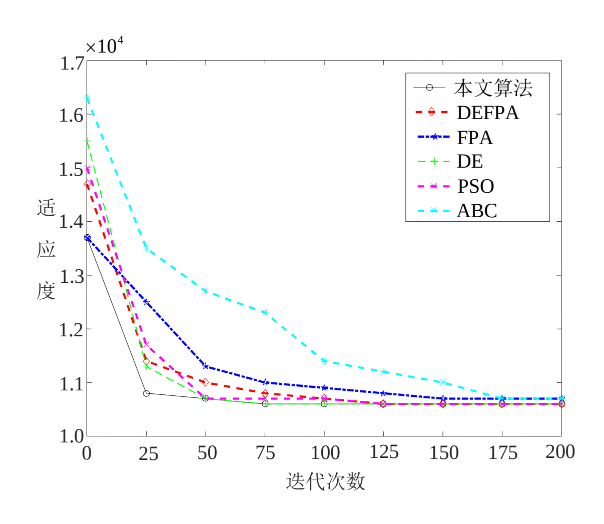


图 6　各算法在 Heart 上的收敛图

从以上的实验中可以发现，本文算法在 5 个数据集上的最优值、最差值、均值和标准差优于其他算法，显示了新算法具有较强的稳定性和鲁棒性。同时本文算法在迭代初期加入了混沌序列，提高了种群多样性，使得算法收敛速度、全局搜索能力有明显提高；而在后期加入禁忌搜索，可增强算法跳出局部最优解的能力。但对于部分数据集本文算法在迭代初期表现不理想，初始的适应度较高，甚至高于 FPA 算法，这将是进一步优化的部分。而在不同的数据集下 DEFPA、FPA、DE、PSO、ABC 算法表现各有不同，从整体可以看出 PSO 和 DE 算法性能更好，DEFPA 次之，ABC 和 FPA 最差，同时此次实验可说明针对不同的数据集采用适合的算法可得到较好的聚类结果。

4.2　改进算法的聚类实验

在上一节中进行了大量的数值模拟实验，实验结果显示本文算法在解决聚类问题有一定的有效性，同时也表明本文算法是一种收敛速度更快、稳定性更高、可避免局部最优解的算法。在本节中，采用上一节所使用的 6 个算法和 5 个数据集进行聚类实验，通过比较迭代 50 次的平均聚类准确率进一步分析算法性能。

表 7 聚类准确率的平均值

算法	art1	art2	Iris	wine	Heart
本文算法	**97.92**	**96.78**	**96.41**	**93.76**	**76.28**
DEFPA	90.21	91.62	93.78	92.48	71.39
FPA	83.01	84.06	90.16	79.02	63.92
DE	92.25	93.82	94.71	76.25	70.83
PSO	94.39	95.06	93.02	80.24	71.37
ABC	84.61	88.37	92.73	76.41	52.74

由表 7 可以看出，本文算法在 5 个数据集上的聚类准确率均优于其他算法，说明加入混沌序列和禁忌搜索，使得算法全局搜索能力增强，能够有效的跳出局部最优解，提高了聚类精度。相比另 5 个算法，本文算法的聚类准确率有很大提高，在 art1 数据集中比最优的 PSO 算法高出 3.53%，在 art2 数据集中比最优的 PSO 算法高出 1.72%。在 Iris 数据集中比最优的 DE 算法高出 1.7%，在 wine 数据集中比最优的 DEFPA 算法高出 1.28%，在 Heart 数据集中比最优的 DEFPA 算法高出 4.89%，这说明本文算法有较强全局搜索和避免局部最优的能力，能使花粉找到聚类效果最优的解。而 DEFPA 算法在 wine 数据集表现出较好的聚类效果，聚类准确率明显高于 DE、PSO、ABC 算法，FPA 算法对每个数据集聚类效果都不理想，DE 算法对 Iris 数据集聚类效果较好，准确率可达到 84.71%，PSO 算法对 art1 数据集聚类能力最好，高于 FPA 算法 11.38%，ABC 算法的聚类结果与 FPA 相近。

5　结束语

本文首先提出了一种改进的花朵授粉算法，分别在初始阶段加入混沌序列和在后期引入禁忌搜索，以解决原始算法随机初始化和易陷入局部最优解的问题。随后，针对 k-means 算法前期易受初始簇类中心的影响，在数据聚类中会导致聚类结果不精确和不稳定，结合改进的花朵授粉算法和 k-means 聚类，提出了一种能精准确定聚类中心的聚类算法。实验结果表明本文算法可提高聚类效果，加快寻优能力，同时避免了局部最优的问题。但本文算法在时间复杂度上表现较差，初始适应度较高，对于高维数据集聚类效果较差，这些将是下一步研究内容。

参考文献：

[1] Jain A K. Data clustering: 50 years beyond K-means [J]. Pattern Recognition Letters, 2010, 31 (8): 651-666.

[2] Duwairi R, Abu M. A novel approach for initializing the spherical K-means clustering algorithm [J]. Simulation Modeling Practice and Theory, 2015, 54 (5): 49-63.

[3] Kumar K M, Reddy A R M. An efficient k-means clustering filtering algorithm using density based initial cluster centers [J]. Information Sciences, 2017, 418: 286-301.

[4] Bianchi F M, Livi L, Rizzi A. Two density-based k-means initialization algorithms for non-metric data clustering [J]. Pattern Analysis and Application, 2016, 19 (3): 1-19.

[5] Hu Jixiong, Wang Chunzhi, Liu Chuan, *et al.* Improved K-means algorithm based on hybrid fruit fly optimization and differential evolution [C]// Proc of the 12th International Conference on Computer Science and Education. Piscataway, NJ: IEEE Press, 2017: 464-467.

[6] Rahman M A, Islam M Z. A hybrid clustering technique combining a novel genetic algorithm with K-means [J]. Knowledge-Based Systems, 2014, 71 (71): 345-365.

[7] 王波，余相君. 自适应布谷鸟搜索的并行 K-means 聚类算法 [J]. 计算机应用研究, 2018, 35 (3): 675-679. (Wang Bo, Yu Xiangjun. Parallel K-means clustering algorithm based on adaptive cuckoo search [J]. Application Research of Computers, 2018, 35 (3): 675-679.)

[8] Yang Xinshe. Flower pollination algorithm for global optimization [C]// Proc of Unconventional Computing and Natural Computation. Berlin: Springer, 2012: 240-249.

[9] Draa A. On the performances of the flower pollination algorithm–Qualitative and quantitative analyses [J]. Applied Soft Computing, 2015, 34 (C): 349-371.

[10] Sayed S A, Nabil E, Badr A. A binary clonal flower pollination algorithm for feature selection [J]. Pattern Recognition Letters, 2016, 77 (C): 21-27.

[11] Galvez J, Cuevas E, Avalos O. Flower pollination algorithm for multimodal optimization [J]. International Journal of Computation-al Intelligence Systems, 2017, 10 (2107): 627-646.

[12] 肖辉辉，万常选，段艳明. 一种改进的新型元启发式花朵授粉算法 [J]. 计算机应用研究, 2016, 33 (1): 127-131. (Xiao Huihui, Wan Chang-xuan, Duan Yanming. Improved novel metaheuristic flower pollination algorithm [J]. Application Research of Computers, 2016, 33 (1): 127-131.)

[13] 肖辉辉. 基于单纯形法和自适应步长的花朵授粉算法 [J]. 计算机工程与科学, 2016, 38 (10): 2126-2133. (Xiao Huihui. A flower pollination algorithm based on simplex method and sel-adaptive step [J]. Computer Engineering and Science, 2016, 38 (10): 2126-2133.)

[14] 肖辉辉，万常选，段艳明，等. 融合高斯变异和 Powell 法的花朵授粉优化算法 [J]. 计算机科学与探索, 2017, 11 (3): 478-489. (Xiao Huihui, Wan Changxuan, Duan Yanming, *et al.* Flower pollination algorithm combination with gauss mutation and Powell search method [J]. Journal of Frontiers of Computer and Technology, 2017, 11 (3): 478-489.)

[15] 吴秀丽，周永权. 一种基于混沌和单纯形法的水波纹优化算法 [J]. 计算机科学, 2017, 44 (5): 218-225. (Wu Xiuli, Zhou Yongquan. Improved water wave optimization algorithm based on chaos optimization and simplex method [J]. Computer Science, 2017, 44 (5): 218-225.)

[16] Karimi E, Maleki H, Reza A. Tabu search algorithm to solve the intermodal terminal location problem [J]. Journal of Mathematical Extension, 2015, 9 (1): 75-89.

[17] Niknama T, Amiri B. An efficient hybrid approach based on PSO ACO and k-means for cluster analysis [J]. Applied Soft Computing, 2010, 10 (1): 183-197.

[18] Blake C L, Merz C J. UCI repository of machine learning databases [DB/OL]. [2018-04-29] http://archive. ics. uci. edu/ml/datasets. html.